

Toward Building Video Multicast Tree with Congestion Avoidance Capability in Software-Defined Networks

Huifen Huang¹, Zhihong Wu², Jin Ge³, and Lu Wang³

¹School of Information Science and Electrical Engineering, ShanDong JiaoTong University, China

²Qilu Hospital, Shandong University, China

³Shandong Provincial Key Laboratory of Computer Networks, SCSC, China

Abstract: Network congestion is an obstacle to a Quality of Service (QoS) guarantee for online video applications, because it leads to a high packet loss rate and long transmission delay. In the Software-Defined Network (SDN), the controller can conveniently obtain the network topology and link bandwidth use situation. Based on the above advantages, an SDN-based video multicast routing solution, called Congestion Avoidance Video Multicast (CAVM), is proposed in this paper. CAVM obtains overall network topology, monitors available bandwidth resource and measures the link delays based on the OpenFlow, a popular SDN southbound interface. We introduce a novel multicast routing problem, named the Delay-Constrained and Minimum Congestion-Cost Multicast Routing (DCMCCMR) problem, which finds the multicast tree with the lowest congestion cost and a source-destination delay constraint in the SDN environment. The DCMCCMR problem is NP-hard. CAVM uses an algorithm to solve it in polynomial time. Our experimental results confirm that the proposed algorithm can build multicast trees with good congestion avoidance capability.

Keywords: Network congestion, multicast, Software-Defined Network, congestion avoidance.

Received May 21, 2017; accepted May 7, 2018

<https://doi.org/10.34028/iajit/17/2/3>

1. Introduction

Currently, online video dominates the world's Internet traffic. A well-known method for alleviating the traffic of online video applications is to use Internet Protocol (IP) multicast as the communication method, because only one data copy passes through any of its related links [2]. Existing multicast protocols are mainly based on two types of forwarding structures: Shortest Path Tree (SPT) and Minimum-cost Spanning Tree (MST). The SPT-based multicast protocols (e.g., Protocol Independent Multicast-Sparse Mode (PIM-SM) [16]) attempt to minimize the distance from the sender to each receiver, whereas the MST-based multicast protocols (e.g., Delay-Constrained Minimum Path Heuristic (DCMPH) [30]) aim at minimizing the overall cost of the multicast tree. Network congestion can lead to a high packet loss rate and long transmission delay. Therefore, it is an obstacle to a Quality of Service (QoS) guarantee for some video multicast applications such as video conferencing and remote consultation.

Multicast congestion control has been researched for a long time. Existing solutions usually reduce the data sending rate to avoid network congestion. Special video coding (e.g., layered coding [27]) can also alleviate the network congestion problem by reducing the video quality. The multicast tree with good

congestion avoidance capability can improve the multicast QoS to a greater degree. However, it is difficult to obtain required network conditions, including network topology and available bandwidth resources, in the current Internet architecture. As a result, existing multicast protocols, including SPT-based and MST-based multicast protocols, usually do not consider to optimize the congestion avoidance capability.

The Software-Defined Network (SDN) is a new network paradigm, which separates the data plane and control plane. In the SDN, a logically centralized controller manipulates the switches through the southbound interface, and switches simply forward the packets according to the flow entries planned by the controller. Most existing SDN controllers use openflow [11] as a southbound interface. With its advantages (e.g., full knowledge of the network topology, convenient traffic monitoring and logically centralized control), the SDN brings new power to the multicast technique.

In this paper, we introduce an SDN-based solution, named Congestion Avoidance Video Multicast (CAVM), for building a video multicast tree with good congestion avoidance capability. CAVM is designed to offer a high-stability multicast forwarding structure for congestion-sensitive multicast applications such as online video conferencing and remote consultation.

Our proposed solution employs openflow-enabled methods to obtain the overall network topology, monitor available bandwidth resources, and measure the link delays. We introduce metrics for evaluating the congestion cost of a multicast path and further propose a Delay-Constrained and Minimum Congestion-Cost Multicast Routing (DCMCCMR) problem, which finds the multicast tree with the least path congestion cost and a source-destination delay constraint.

The remainder of this paper is organized as follows. We introduce related research in section 2. Section 3 introduces the system architecture for our proposed solution. We explain how to build a SDN-based multicast tree with good congestion avoidance capability in section 4. The results of the performance evaluation are presented in section 5. Finally, we give our conclusions in section 6.

2. Related Work

The SDN is an innovative network paradigm, which decouples the control plane and data plane. In a traditional network, the multicast trees are usually built using a flood-and-prune model, e.g., Protocol Independent Multicast--Sparse Mode (PIM-SM) [16], Distance Vector Multicast Routing Protocol (DVMRP) [23], and Multicast Open Shortest Path First (MOSPF) [12]. In contrast to the traditional network, the SDN centrally finds the multicast trees based on global topology information. In addition, the SDN facilitates immediate deployment and the flexible adaptability of multicast because of its programmable nature [14]. Multimedia group applications sometimes involve a large number of concurrent users, which might incur a high load on the SDN controller(s). The scalability of control is a concern for the SDN technique, which has been studied widely (e.g., [26]). Kreutz *et al.* [8], it was shown that the flow handling capabilities of some controller platforms (e.g., Maple [22]) exceed 10 M flows per second. In this study, we assume that SDN control can satisfy the multicast requirements.

Some previous studies have considered multicast techniques in the SDN environment. Huang *et al.* [5] introduced a multicast tree for SDN called the branch-aware Steiner tree, which minimizes the sum of the number of tree edges and the number of tree branch nodes. Sheu *et al.* [19] proposed two multicast algorithms for solving the multicast problem in an SDN environment. Both of these algorithms consider the balance between bandwidth utilization and the source-destination delay. Hasrouty *et al.* [4] proposed an algorithm for building a minimum spanning tree or a shortest path tree. Jiang *et al.* [7] extended Dijkstra's well-known shortest path algorithm by simultaneously considering the edge weights and node weights for a graph derived from the underlying SDN topology. Several packet loss recovery solutions (e.g., [10, 28])

for the SDN-based multicast have been proposed. Zhang *et al.* [29] proposed a video multicast orchestration scheme based on SDN for 5G Ultra-dense networks. The above multicast techniques do not consider the multicast congestion avoidance problem.

Existing multicast congestion control solutions usually reduce the data sending rate to avoid network congestion, e.g., TCP-Friendly Multicast Congestion Control (TFMCC) [25], Smooth Multicast Congestion Control (SMCC) [9] and Pragmatic General Multicast Congestion Control (PGMCC) [18]. TFMCC is a single rate congestion control mechanism for multicast transmissions in a best-effort Internet environment. In TFMCC, the sending rate is adapted to the receiver experiencing the worst network conditions. SMCC is a multiple rate equation-based congestion control algorithm for layered multicast sessions. PGMCC is also a single rate multicast congestion control scheme which is TCP-friendly and can achieve scalability, stability and fast response to variations in network conditions. Unlike the above work, we strive for building a multicast tree with good congestion avoidance capability.

3. System Architecture

In this section, we describe the architecture of our proposed video multicast routing solution, i.e., CAVM.

As illustrated in Figure 1, CAVM consists of the following components: topology monitor, used to obtain the network topology and its updates; delay measurement, used to measure link delays; traffic monitor, used to obtain the maximum and current transmission speeds of switch ports; video multicast routing, used to compute the video multicast tree aimed at optimizing the congestion avoidance capability. In addition, CAVM employs the following two tables: the available bandwidth table, used to store the available bandwidth of physical links; the link delay table, used to store the transmission delays of physical links. CAVM can be integrated into existing controllers supporting the openflow specification.

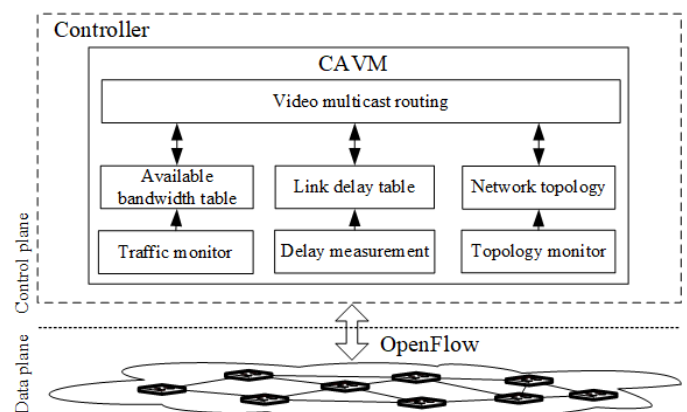


Figure 1. CAVM architecture.

Previous research studies have shown that it is feasible to monitor and measure the openflow network [1, 13, 15]. CAVM uses existing methods to implement related monitoring and measurement. In an SDN, it is easy to obtain the network topology, and we do not present a further explanation. In the openflow specification, using version 1.3.4 as the sample, OFPMP_PORT_DESCRIPTION denotes the port description request that enables the controller to obtain a description of all the standard ports of an openflow switch. The reply body includes the fields curr-speed and max-speed, which denote the current and maximum bitrates of the port, respectively. Similarly to the flow status sampling in PayLess [1], CAVM queries the port status based on a dynamic period. The port status query period ranges from τ_1 to τ_2 ($\tau_1 < \tau_2$), where τ_1 and τ_2 are two configuration parameters. Initially, the port status query period is set to τ_2 . The SDN controller sends a port description request to each switch. When the controller receives the reply to a port description request, it compares the returned curr-speed values with the corresponding values obtained in the previous query procedure. If the curr-speed value of at least one port of the switch A has clearly changed, the port status query period of A is set to $\min\{\tau_1, \lambda - \Delta\}$, where λ denotes the current value of A's port status query period and Δ indicates a variation value; otherwise, the port status query period of A is set to $\max\{\tau_2, \lambda + \Delta\}$. We use the above dynamical port status query period adjustment to attempt to obtain a tradeoff between the accuracy and the overhead of the traffic measurement. Assume that:

1. A link connects the port n of the switch S_1 and the port m of the switch S_2 .
2. The maximum and current bitrates of S_1 : n are x and y, respectively.
3. The maximum bitrates of S_2 : m is z; then, the available bandwidth of the directed link $S_1:n \rightarrow S_2:m$ is $\frac{\min\{x,z\}-y}{8}B/s$. Phemius and Bouet presented a link delay measurement method, which has high accuracy and a low footprint [15]. CAVM uses this method to measure the link delay.

In CAVM, the video multicast source sends a register message, including the transport port and bandwidth requirement, to the controller. The $\langle IP \text{ address, transport port} \rangle$ is used to identify a multicast session. When receivers join a video multicast session, the access switches send PackIn messages to the controller. Then, the controller computes delay-constrained and congestion-evasive routes for these receivers, based on the information on available bandwidth, link delay, and network topology. The controller sends flow installation instructions to the related switches in terms of the computed routes. CAVM uses a special media server to manage a multicast group. If an inactive receiver is detected, the

controller sends OFPFC_DELETE messages to the related switches to remove unnecessary flows.

4. Multicast Tree Construction

4.1. Problem Description

We model a network as a directed graph $N=(V,E)$, where V is the set of vertices that represent communication nodes (switches and hosts) and E is the set of edges that denote communication links. We describe the dynamic multicast group using a 3-tuple (s, R_e, R_n) , where s denotes the multicast source, R_e the set of existing receivers, and R_n the set of new receivers. Let $\omega(l)$ indicate the use frequency of a link l . As shown in Payless [1], the SDN technique enables the controller to monitor the per-flow status. Therefore, it might be feasible to define $\omega(l)$ in terms of the historical data and knowledge discovery technique. In this study, we consider a structure-based definition to approximatively evaluate the link use frequency with low cost. Let $V_a (V_a \subset V)$ denote the set of access switches and $E' (E' \subset E)$ denote the links that connect two different switches. For the link k that connects a host and an access switch, we define $\omega(k) = 0$. For the link l that connects two different switches, we define $\omega(l)$ as

$$\omega(l) = \frac{\sum_{i \in V_a} \sum_{j \in V_a} X_{i,j,l}}{\max\{\sum_{i \in V_a} \sum_{j \in V_a} X_{i,j,k} | k \in E'\}} \quad (1)$$

Where

$$X_{i,j,l} = \begin{cases} 1 & \text{if } l \text{ is in the shortest path from } i \text{ to } j \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

We define $f(l, s, R_e)$ to evaluate the congestion degree of a link l associated with the multicast source s and a set R_e . Let B_M denote the maximum available bandwidth of the links, b_l denote the available bandwidth of the link l , and B_s denote the required bandwidth of the multicast session sourced from s . Then the function $f(l, s, R_e)$ is defined as

$$f(l, s, R_e) = \begin{cases} 0 & \text{if } l \in L_{R_e}, \\ \omega(l) \frac{B_M - b_l + B_s}{B_M} & \text{if } l \notin L_{R_e}, b_l \geq B_s, \\ \omega(l)(B_s - b_l) & \text{if } l \notin L_{R_e}, b_l < B_s, \end{cases} \quad (3)$$

Where L_{R_e} denotes the set of the links through which the existing multicast flows, for the receivers of R_e , pass. The links are selected based on congestion degrees in the multicast routing procedure. According to the above definition, we can observe that the links through which the multicast flow has passed are selected with the highest priority, and that the links, which have no or little available bandwidth, are selected with low priority. Thus, we consider that this definition facilitates the improvement of the congestion avoidance capability.

We introduce DCMCCMR, which finds the multicast paths for new receivers that satisfy the following requirements:

$$\text{Minimize: } \sum_{\xi \in P(R_n)} h(\xi) \sum_{k \in \xi} f(k, s, R_e) \text{ subject to} \\ \forall(k) \in R_n, sdc(s, k) \leq D_s \quad (4)$$

Where D_s denotes the delay constraint, $sdc(s, k)$ denotes the sum of the delays of the links in the path from s to k , $P(R_n)$ denotes the set of the multicast paths (i.e., source-destination paths) for R_n , and $h(\xi)$ denotes the number of links in a multicast path ξ . Note that the multicast paths for the receivers compose the multicast tree. From the definition, we can see that the DCMCCMR problem optimizes every multicast path. A multicast path is congested if there exists at least one congested link in it. Therefore a multicast path is easier to be subject to congestion if it contains more links. In addition, a multicast path is also easier to be subject to congestion if the total congestion degree along this path is higher. Therefore the DCMCCMR problem minimizes the product of the link number and the total congestion degree for each multicast path. In this paper, the product of the link number and the total congestion degree for a multicast path is said to be the congestion cost of this path. The Weight-Constrained Shortest Path Problem (WCSPP) consists of finding a least-cost path between two nodes, such that the total weight along the path is less than a specified value. The WCSPP problem is NP-hard [3]. Assume that the links have the same congestion degree, then the DCMCCMR problem is transformed to a WCSPP problem. Thus, the DCMCCMR problem is NP-hard.

As mentioned previously, existing multicast routing protocols are mainly based on SPT and MST. The DCMCCMR problem is different from MST-based multicast routing problems because the DCMCCMR problem aims at optimizing each multicast path but MST-based multicast routing problems attempt to optimize the overall tree cost. SPT-based multicast routing problems strive for minimizing the distance between the multicast sender and each receiver. As opposed to SPT-based multicast routing problems, the DCMCCMR problem attempts to minimize a product of two different metrics of each multicast path.

4.2. Algorithm

CAVM is designed to compute the multicast routes for a group of new receivers, which indicates that it simultaneously supports the static and dynamic multicast tree construction models. Algorithm 1 describes the multicast routing for new receivers. This algorithm uses Floyd-Warshall algorithm to compute the shortest switch-to-switch paths. Note that the Floyd-Warshall algorithm is a simple and widely used algorithm to compute shortest paths between all pairs of vertices. Algorithm 1 uses three types of link cost: delay, congestion degree, and bandwidth-restricted delay. If the link has sufficient available bandwidth for a designated multicast session, then its bandwidth-restricted delay is equal to the link delay;

otherwise, its bandwidth-restricted delay is infinite. We use N_d , N_c , and N_b to denote the network graphs using delay, congestion degree, and bandwidth-restricted delay as the metrics of the link cost, respectively. The notation $dl(\xi)$ denotes the sum of the delays of the links in the path ξ , $h(\xi)$ denotes the number of the links in the path ξ , $cl(\xi)$ denotes the sum of the congestion degrees of the links in the path ξ , $bsp(x, y)$ denotes the shortest path from x to y in N_b , $sp(x, y)$ denotes the shortest path from x to y in N_d , and $pa(\xi, x, y)$ denotes the path from x to y in ξ . Because the switches are relatively steady, the shortest switch-to-switch paths in N_d are computed in advance.

As mentioned previously, the DCMCCMR problem is NP-hard. Algorithm 1 solves the DCMCCMR problem using the following four successive steps:

1. Compute the multicast path, with minimum total congestion degree, for a designated new receiver.
2. If the above step cannot obtain a delay-constrained path, then reshape the path computed in the first step by replacing some path segmentations with shorter paths without congestion.
3. If the above two steps cannot obtain a delay-constrained path, then reshape the path computed in the first step by replacing some path segmentations by the shortest paths.

We can notice that Algorithm 1 can obtain a multicast path, without congestion, for the receiver if there exists at least one path that is not congested and satisfies the delay constraint. It is impossible to ensure that the collected information on link traffic is in complete accordance with the current status. However, the proposed algorithm can effectively alleviate the negative effect of the information inconsistency, because it reduces the congestion as much as possible.

It takes $O(|V|^3)$ to compute the shortest switch-to-switch paths in N_b . The loop starting from "For $r \in R_n$ do" takes at most $O(|R_n||V|^2)$. Therefore, the complexity of Algorithm 1 in the worst case is $O(|V|^3 + |R_n||V|^2)$. As noted previously, the shortest switch-to-switch paths can be computed in advance because the network topology is relatively steady. In the above case, the complexity of Algorithm 1 in the worst case is $O(|R_n||V|^2)$, which indicates that Algorithm 1 has a slightly higher complexity than the common shortest-path tree construction that uses the Dijkstra algorithm. Algorithm 1 is designed to offer desirable congestion avoidance capability under a delay constraint, which is more complicated than the algorithm that only attempts to compute shortest paths. Therefore we say that the complexity of Algorithm 1 is acceptable.

Algorithm 1: Multicast routing algorithm

```

# s: multicast source; Rn: the set of new receivers
Compute the congestion degree for each link.
Use the Floyd-Warshall algorithm to compute the
shortest switch-to-switch paths in Nb.
For each (r in Rn)
{
  Compute the shortest paths from s to other nodes in
  Nc using the Dijkstra algorithm.
  ξ, ξ' ← the shortest path from s to r.
  If dl(ξ) > Ds then
  # The following code reshapes the path ξ.
  conti ← true; c ← 2
  While dl(ξ) > Ds and conti == true do
    #Let the nodes in ξ be denoted by
    μ1 = s, μ2, ..., μm = r.
    If m ≤ c then
      ξ ← bsp(s, r); conti ← false.
    Else
      maxg ← 0; k ← 0.
      for (i = 1 to m - c)
      {
        g ←  $\frac{dl(pa(\xi, u_i, u_{i+c})) - dl(bsp(u_i, u_{i+c}))}{cl(bsp(u_i, u_{i+c})) - cl(pa(\xi, u_i, u_{i+c}))}$ 
        If g > maxg then
          maxg ← g; k ← i.
        end If
      }
      If maxg > 0 then
        Replacing the path from uk to uk+c in ξ by
        bsp(uk, uk+c)
      Else
        c ← c + 1.
      end If
      end If
      end While
      end If
      If dl(ξ) ≤ Ds then
        The congestion cost of each link in ξ is set to 0.
      Else
        ξ ← ξ'.
        Reshape ξ according to the above path reshaping
        procedure, except that bsp(x, y) is replaced by sp(x, y).
      end If
      ξ is the multicast path for r.
    }
  }

```

4.3. Example

We further explain the congestion avoidance capability of the proposed algorithm through an example shown in Figure 2. In the topology shown in Figure 2, the delay of each link connecting a video source or receiver is 2 ms; the consumed bandwidth and total bandwidth of each link connecting a video source or receiver are 1 Mbps and 10 Mbps, respectively. The bandwidth requirement and delay constraint of the multicast group in Figure 2 is 1 Mbps and 100 ms. For the sake of simplicity, we assume that the two directed links corresponding to a physical link have the same delay and bandwidth.

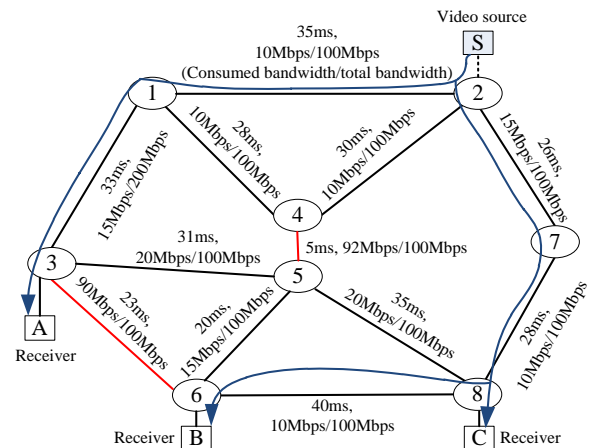


Figure 2. An application example of CAVM.

We can note that the shortest path, using delay as the metrics, from the video source to receiver A is $S \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow A$. However, the sum of congestion degree, defined by Equation (3), of the above path is larger than that of the path $S \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow A$. According to the proposed algorithm, $S \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow A$ is selected as the multicast path for A. Similarly, $S \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 6 \rightarrow B$ is selected as the multicast path for B, and $S \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow C$ is selected as the multicast path for C. We can notice that the proposed algorithm effectively avoids the links $3 \rightarrow 6$ and $4 \rightarrow 5$, which are close to congestion.

5. Results and Discussion

We generated a 6000-node Waxman graph [24], which is a common model that represents the networks, as the network topology. The nodes in the Waxman graph were used as the routing nodes. In the SDN environment, both openflow-enable switch and openflow-enable router forward received packets according to matched flow entry. Therefore a routing node denotes either a router or a switch in the SDN environment. The multicast source and receivers were randomly connected to routing nodes in the Waxman graph. The delay and bandwidth of each link, which connected a routing node and a receiver (or the multicast source) was 2 ms and 10 Mbps, respectively.

Note that the delay and bandwidth of each link, which connected two routing nodes, were automatically configured when generating the Waxman graph. We use the Bandwidth Consumption Rate (BCR) to denote the ratio of the used bandwidth of a link to the total bandwidth of this link. The initial BCRs of the links were assigned a random value ranging from 0.1 to 0.9 to reflect changes of link background traffic; the source-destination delay constraint was 1000ms. The multicast bandwidth requirements ranged between 0.2Mbps to 2.7Mbps, which referred to the video streaming rate in the dataset YouMod introduced in [17]. We developed a program to build related multicast trees based on generated network conditions including network topology and link information. The correctness of the program was verified by manually checking the results of many instances including 10-20 receivers. The data in the graphs and the table introduced in next parts represent average values of 100 runs.

We evaluated the congestion avoidance capability of CAVM in comparison with the DCMPH [30], MA [21] and LCA [20]. Note that DCMPH and MA strive for building weight-constrained MSTs, while LCA attempts to build weight-constrained SPT. DCMPH, MA and LCA do not try to optimize the congestion avoidance capability. Therefore they do not introduce how to evaluate the congestion avoidance capability. For fairness, all the multicast trees were built using the value computed by Equation (3) as the link cost (or distance) and using the delay as an additional link weight.

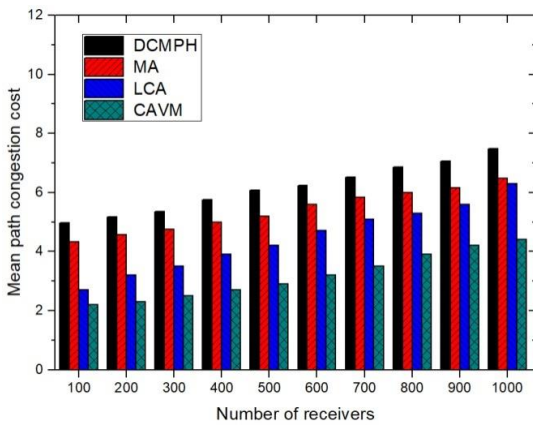


Figure 3. Comparison of mean path congestion cost among CAVM, DCMPH, MA and LCA.

Figures 3 and 4 present the comparisons of mean and maximum path congestion costs in CAVM, DCMPH, MA and LCA. In the two figures, it can be seen that both mean and maximum path congestion costs in CAVM are clearly lower than that in DCMPH, MA and LCA. The lower the path congestion cost is, the stronger the congestion avoidance capability of the corresponding multicast path. Therefore we say that CAVM significantly improves the congestion avoidance capability.

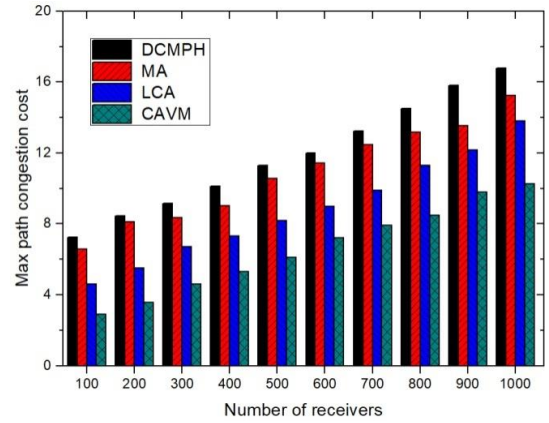


Figure 4. Comparison of mean path congestion cost among CAVM, DCMPH, MA and LCA.

Figure 5 presents a comparison of the source-destination delay of CAVM, DCMPH, MA and LCA. In our experiments, the CAVM tree had no advantage on reducing mean source-destination delay. The main reason is that the main objective of CAVM is to improve the congestion avoidance capability.

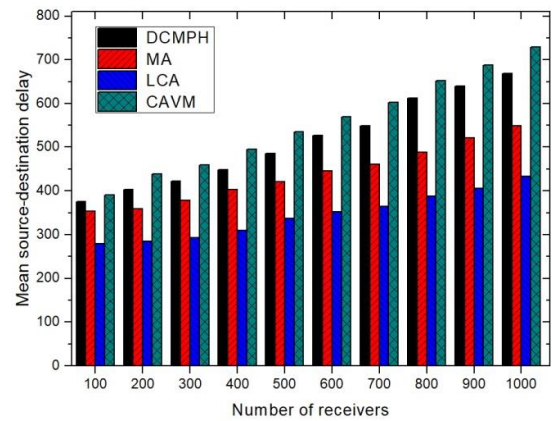


Figure 5. Comparison of source-destination delay among CAVM, DCMPH, MA and LCA.

We also evaluated the congestion avoidance capability of CAVM based on an 81-node topology described by the Ulaknet dataset in the Internet Topology Zoo [6]. The datasets in the Zoo topologies include no link delay information. To cope with the above situation, the delay constraint was not considered in our experiment. Other parameters, such as the BCR range, multicast bandwidth requirement and the bandwidths of links connecting receivers, were configured as the methods introduced previously. Table 1 presents the experimental results. The results show that CAVM has better network congestion avoidance capability than other solutions.

Table 1. Comparison of path congestion cost based on the Ulaknet topology.

Solutions	10 receivers		20 receivers		30 receivers	
	Avg.	Max	Avg.	Max	Avg.	Max
CAVM	2.5	3.8	3.4	4.3	4.2	5.1
DCMPH	3.6	4.3	4.9	5.6	5.8	6.5
MA	3.2	3.7	4.6	5.2	5.4	6.1
LCA	2.7	3.5	3.9	4.9	4.7	5.6

6. Conclusions

In this paper, we have proposed an SDN-based multicast routing solution for online video applications. Our solution monitors the network and measures the link delays based on the openflow specification. In this paper, a function to evaluate the congestion degree was defined, and the DCMCCMR problem was introduced. This DCMCCMR problem is NP-hard. We proposed an algorithm to approximately solve it in polynomial time. Our solution can build a video multicast tree with good congestion avoidance capability, which facilitates the assurance of the video QoS. In future work, we plan to extend our solution to include the capability to rearrange some multicast paths when they are congested because of subsequent traffic. A further plan is to study link use frequency evaluation method based on the knowledge discovery.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 61472230, and the Shandong provincial Natural Science Foundation of China under Grant No. ZR2015JL023, and Shandong Soft Science Planning Item of China under Grant No. 2016RZB01035.

References

- [1] Chowdhury S., Bari M., Ahmed R., and Boutaba R., "Payless: A Low Cost Network Monitoring Framework for Software Defined Networks," in *Proceedings of IEEE Network Operations and Management Symposium*, Krakow, pp. 1-9, 2014.
- [2] Derdouri L., Pham C., and Mohamed M., "Analysis of Hybrid Router-Assisted Reliable Multicast Protocols in Lossy Networks," *The International Arab Journal of Information Technology*, vol. 14, no. 1, pp. 26-34, 2017.
- [3] Dumitrescu I. and Boland N., "Algorithms for the Weight Constrained Shortest Path Problem," *International Transactions in Operational Research*, vol. 8, no. 1, pp. 15-29, 2002.
- [4] Hasrouly C., Autefage V., Olariu C., Magoni D., and Murphy J., "Sdn-Driven Multicast Streams with Adaptive Bitrates for Voip Conferences," in *Proceedings of IEEE International Conference on Communications*, Kuala Lumpur, pp. 1-7, 2016.
- [5] Huang L., Hung H., Lin C., and Yang D., "Scalable steiner Tree for Multicast Communications in Software-Defined Networking", Available at: <http://arxiv.org/abs/1404.3454>, Last Visited, 2018.
- [6] Internet Topology Zoo, Available at: <http://www.topology-zoo.org/dataset.html>, Last Visited, 2018.
- [7] Jiang J., Huang H., Liao J., and Chen S., "Extending Dijkstra's Shortest Path Algorithm for Software Defined Networking," in *Proceedings of 16th Asia-Pacific Network Operations and Management Symposium*, Hsinchu, pp. 1-4, 2014.
- [8] Kreutz D., Ramos F., Verissimo P., Rothenberg C., Azodolmolky S., and Uhlig S., "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14-76, 2015.
- [9] Kwon G. and Byers J., "Smooth Multirate Multicast Congestion Control," in *Proceedings 22nd Annual Joint Conference of the IEEE Computer and Communications*, pp. 1022-1032, 2003.
- [10] Mahajan K., Sharma D., and Mann V., "Athena: Reliable Multicast for Group Communication in SDN-Based Data Centers," in *Proceedings of 9th International Conference on Communication Systems and Networks*, Bangalore, pp. 174-181, 2017.
- [11] McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., and Turner J., "Openflow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69-74, 2008.
- [12] Moy J., "Multicast Extensions to Ospf," IETF RFC 1584, Network Working Group, 1994.
- [13] Niels L., Adrichem M., Doerr C., and Kuipers F., "Opennetmon: Network Monitoring in Openflow Software-Defined Networks," in *Proceedings of IEEE Network Operations and Management Symposium*, Krakow, pp. 1-8, 2014.
- [14] Noghani K. and Sunay M., "Streaming Multicast Video Over Software-Defined Networks," in *Proceedings of IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, Philadelphia, pp. 551-556, 2014.
- [15] Phemius K. and Bouet M., "Monitoring Latency with Openflow," in *Proceedings of 9th International Conference on Network and Service Management*, Zurich, pp. 122-125, 2013.
- [16] Pusateri T., "Protocol Independent Multicast-Sparse Mode," IETF RFC 4602, Network Working Group, 2006.
- [17] Rao A., Legout A., Lim Y., Towsley D., Barakat C., and Dabbous W., "Network Characteristics of Video Streaming Traffic," in *Proceedings of the 17th Conference on Emerging Networking Experiments and Technologies*, Tokyo, pp. 1-12, 2011.
- [18] Rizzo L., "Pgmcc: A TCP-Friendly Single-Rate Multicast Congestion Control Scheme," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 17-28, 2000.

- [19] Sheu J., Chang C., and Chang Y., "Efficient Multicast Algorithms for Scalable Video Coding in Software-Defined Networking," in *Proceedings of Personal, Indoor, and Mobile Radio Communications*, Hong Kong, pp. 2089-2093, 2015.
- [20] Smith O., Boland N., and Waterer H., "Solving Shortest Path Problems with A Weight Constraint and Replenishment Arcs," *Computers and Operations Research*, vol. 39, no. 5, pp. 964-984, 2012.
- [21] Sun M. and Wang H., "The Memetic Algorithm for the Minimum Spanning Tree Problem with Degree and Delay Constraints," in *Proceedings of 15th International Conference on Advanced Communication Technology*, Pyeong Chang, pp. 78-82, 2013.
- [22] Voellmy A., Wang J., Yang Y., Ford B., and Hudak P., "Maple: Simplifying SDN Programming Using Algorithmic Policies," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 87-98, 2013.
- [23] Waitzman D., Partridge C., and Deering S., "Distance Vector Multicast Routing Protocol," IETF RFC 1075, 1998.
- [24] Waxman B., "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617-1622, 1988.
- [25] Widmer J. and Handley M., "TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification," IETF RFC 4654, Network Working Group, 2006.
- [26] Yu M., Rexford J., Freedman M., and Wang J., "Scalable Flow-Based Networking with Difane," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 351-362, 2011.
- [27] Zhang Q., Guo Q., Ni Q., Zhu W., and Zhang Y., "Sender-Adaptive and Receiver-Driven Layered Multicast for Scalable Video over the Internet," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 4, pp. 482-495, 2005.
- [28] Zhang X., Yang M., Wang L., and Sun M., "An OpenFlow-Enabled Elastic Loss Recovery Solution for Reliable Multicast," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1945-1956, 2018.
- [29] Zhang X., Yang M., Zhao Y., Zhang J., and Ge J., "An SDN-Based Video Multicast Orchestration Scheme for 5G Ultra-Dense Networks," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 77-83, 2017.
- [30] Zhou L. and Sun Y., "Delay-Constrained Steiner Tree Algorithm Using mph," *Journal of Computer Re-search and Development*, vol. 45, no. 5, pp. 810-816, 2008.



Huifen Huang received her M.S degree from Shandong University in 2006, and her Ph.D degree from Wuhan University of Technology in 2012. Presently, she is a professor at Shandong Yingcai University, and the director of Computer Application Technology Research Institution of Information Technology College. Her research interests include networking and information security.



Zhihong Wu received her M.S degree from Shandong University in 2005, and her Ph.D degree from Shandong University in 2013. Presently, she is an associate professor at Shandong University. Her current research interests include medical information. She is the corresponding author of this paper.



Jing Ge received her B.E. and Ph.D. degrees from Shandong University in 2010 and 2015, respectively. Now, she is a researcher with the Shandong Computer Science Center (National Supercomputer Center in Jinan). Her research interests mainly include software defined networking and multimedia.



Lu Wang received the B.S. and M.S. degrees from Shandong University, China, in 2004 and 2007, respectively. He is currently a researcher with the Shandong Computer Science Center (National Supercomputer Center in Jinan). His research interests include network protocol, protocol test, and cloud computing.